title": "향후 개발 방향 제안 (Refinement & New Features)",

1. "분석 결과 오버레이 토글 기능",

: "AI 분석 결과가 복잡할 경우, 특정 분석 항목(예: 주름, 모공, 색소)만 선택적으로 보거나 숨길 수 있는 기능을 분석 패널에 추가합니다.",

: "분석 패널(`analysisPanel`)에 체크박스들을 추가하고, 각 체크박스의 상태에 따라 `renderAnalysis` 함수에서 해당 항목을 캔버스에 그릴지 말지를 결정하는 로직을 추가합니다."

2. "사진 메타데이터 편집 기능",

: "파일명 분석이 잘못되었거나 정보를 수정해야 할 경우를 대비해, 사용자가 사진의 촬영일, 시술 상태, 각도 등의 정보를 직접 편집할 수 있는 기능을 제공합니다.",

: "사진 목록이나 뷰어 상단에 작은 '편집' 아이콘을 추가하고, 클릭 시 정보를 수정할 수 있는 모달 창을 띄워 Firestore의 해당 사진 문서를 `updateDoc`으로 업데이트합니다."

3. "환자 정보 상세 페이지 및 편집",

: "현재 환자 목록에서는 기본 정보만 보입니다. 환자 이름을 클릭하면 상세 정보(생년월일, 성별, 메모 등)를 조회하고 편집할 수 있는 기능을 추가합니다.",

: "환자 목록 아이템 클릭 시 `selectPatient`를 바로 호출하는 대신, 상세 정보 모달을 먼저 띄우거나 별도의 상세 정보 뷰로 전환하는 로직을 추가할 수 있습니다."

: "아키텍처 및 성능 최적화 (Architecture & Performance)",

4. "이미지 로딩 최적화 (썸네일 도입)",

: "현재 사진 목록(좌측 사이드바)에서 원본 이미지를 불러와 크기만 줄여 표시하고 있어, 사진이 많아지면 로딩 속도가 크게 저하될 수 있습니다. 원본과 별도로 저해상도 썸네일을 생성하고 목록에서는 썸네일만 불러오도록 개선합니다.",

: "Firebase의 'Resize Images' 확장 프로그램을 사용하면, Storage에 원본 이미지가 업로드될 때마다 자동으로 지정된 크기의 썸네일 이미지를 생성해줍니다. 코드에서는 썸네일 URL을 불러와 `photoList`에 표시하고, 메인 뷰어에서만 원본 URL을 사용하도록 수정합니다."

5. "Firestore 쿼리 최적화 및 인덱싱",

: "사진 필터링 시 여러 조건(`where`)을 사용하고 있습니다. 데이터가 많아지면 복합 쿼리의 성능 보장을 위해 Firestore 콘솔에서 '복합 색인(Composite Index)'을 생성해야 합니다.",

: "브라우저 개발자 도구의 콘솔에 나타나는 Firestore 색인 생성 링크를 클릭하여 필요한 복합 색인을 미리 생성해두는 작업이 필요합니다. 이를 통해 서버단에서 필터링이 효율적으로 이루어집니다."

6. "배포 및 운영 (Deployment & Operations)",

1) "Firebase 보안 규칙 강화",

"실제 의료 데이터를 다루기 위해서는 보안이 매우 중요합니다. 현재는 누구나 데이터에 접근할 수 있는 상태일 가능성이 높습니다. 인증된 사용자만 자신의 데이터에 접근할 수 있도록 Firestore 및 Storage의 보안 규칙을 정교하게 설정해야 합니다.",

"Firebase Authentication을 도입하여 사용자 로그인을 구현하고, 보안 규칙에서 `request.auth.uid`를 사용하여 로그인한 사용자의 데이터만 읽고 쓸 수 있도록 규칙을 작성합니다."

2) "에러 핸들링 및 로깅 개선",

"현재 `console.error`와 `alert`로 처리되는 에러를, 실제 운영 환경에서는 사용자가 겪는 문제를 추적할 수 있도록 외부 로깅 서비스와 연동하는 것을 고려할 수 있습니다.",

"Sentry, LogRocket과 같은 클라이언트 측 에러 추적 서비스를 도입하여, 사용자가 모르는 사이에 발생하는 오류까지 개발자가 파악하고 신속하게 대응할 수 있도록 시스템을 구축합니다."